Process for implementing modular multiplication according to the Montgomery method

Publication number: DE69506674T
Publication date: 1999-05-20
Inventor: MONIER GUY (FR)
Applicant: SGS THOMSON MICROELECTRONICS (FR)
Classification:
- International: ***G06F7/72; G06F11/10; G06F17/10; G09C1/00; H03M13/00; H04L9/10;*** **G06F7/60; G06F11/10; G06F17/10; G09C1/00; H03M13/00; H04L9/10;** (IPC1-7): G06F7/72
- European: G06F7/72M
Application number: DE19956006674T 19951026
Priority number(s): FR19940013594 19941108

Also published as:
EP0712071 (A
US5745398 (A
JP8263316 (A)
FR2726667 (A
EP0712071 (B

Abstract not available for DE69506674T
Abstract of corresponding document: **EP0712071**

Three shift registers (10,11,12) which are sub-dividable are preceded by multiplexers (13,14,15) and allow entry for a multiplier (B), modulo (N) and initialising bits (O). The multiplicand (A) is entered through a multiplexer (24) and register (16) which has a series input and parallel output. Two multiplying circuits (19,20) are entered through registers (16,17) and rocking circuits (21,22). The multiplying circuits also receive data through multiplexers (25,26) from subtraction circuits (27,28) and an addition circuit (30). The results of the comparisons are stored in a storage circuit (35).

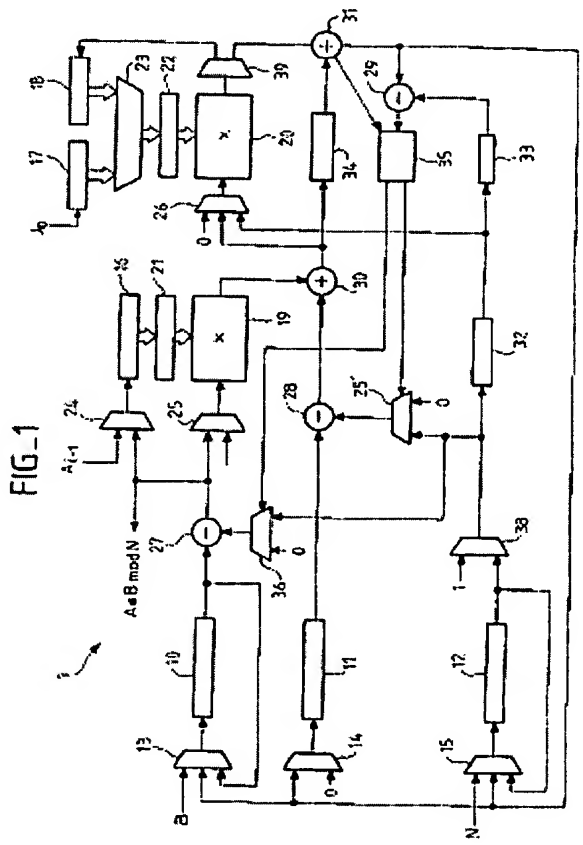

FIG_1

(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENT- UND MARKENAMT

(12) Übersetzung der europäischen Patentschrift

(51) Int. Cl.[6]: G 06 F 7/72

(87) EP 0 712 071 B1

(10) DE 695 06 674 T 2

(21) Deutsches Aktenzeichen: 695 06 674.9
(86) Europäisches Aktenzeichen: 95 470 038.1
(86) Europäischer Anmeldetag: 26. 10. 95
(87) Erstveröffentlichung durch das EPA: 15. 5. 96
(87) Veröffentlichungstag der Patenterteilung beim EPA: 16. 12. 98
(47) Veröffentlichungstag im Patentblatt: 20. 5. 99

(30) Unionspriorität:
9413594 08. 11. 94 FR

(73) Patentinhaber:
SGS-Thomson Microelectronics S.A., Gentilly, FR

(74) Vertreter:
Beetz und Kollegen, 80538 München

(72) Erfinder:
Monier, Guy, c/o Cabinet Ballot Schmit, F-57000 Metz, FR

(84) Benannte Vertragstaaten:
DE, FR, GB, IT

(54) Verfahren zur Verwendung der modularen Multiplikation nach der Montgomery-Methode

DE 695 06 674 T 2

DE 695 06 674 T 2

17.09.96

Die Erfindung betrifft ein Verfahren von Modulo-Multiplikationen nach dem Verfahren von Montgomery. Dieses Verfahren erlaubt Moduloberechnungen auf einem endlichen Körper $GF(2^n)$ (Galois-Körper), ohne Divisionen durchführen zu müssen.

Üblicherweise werden Modulo-Operationen auf $GF(2^n)$ in der Kryptographie bei Anwendungen von Authentifizierung von Nachrichten, Identifizierung eines Benutzer und Austausch von Schlüsseln verwendet. Derartige Anwendungsbeispiele sind z.B. in der französischen Patentanmeldung mit der Veröffentlichungs-Nr. 2 679 054 beschrieben.

Es gibt handelsübliche integrierte Schaltkreise für diese Anwendungen, z.B. von SGS-THOMSON MICROELEKCTRONICS S.A. mit der Bezeichnung ST16CF54, aufgebaut um eine Verbindung aus Zentraleinheit-arithmetischer Coprozessor herum und gedacht für das Umsetzen von Modulo-Berechnungen. Der verwendete Coprozessor erlaubt die Bearbeitung von Modulo-Multiplikationen unter Verwendung des Verfahrens nach Montgomery. Er ist Gegenstand der europäischen Patentanmeldung mit der Veröffentlichungs-Nr. 0 601 907 A2, und er ist dargestellt in Figur 1 (diese Figur entspricht der Figur 2 der genannten europäischen Patentanmeldung).

Die Basisoperation, genannt P_{field}, besteht darin, aus drei binären Daten A (Multiplikand), B (Multiplikator kleiner als N) und N (Modulo), codiert als eine ganze Zahl n von

Bits, eine binäre Dateneinheit $P(A, B)_N$ zu erzeugen, die in n Bit codiert ist, so daß $P(A, B)_N = A \cdot B \cdot I \bmod N$ mit $I = 2^{-n} \bmod N$. Dazu wird angenommen, daß die Daten in m Worten à k Bit mit $n \cdot k = n$ codiert sind und die Worte von A und B einen Multiplikatorschaltkreis mit einem seriellen Eingang, einem parallelen Eingang und einem seriellen Ausgang zugeführt werden.

In den Coprozesser, der in der genannten europäischen Patentanmeldung beschrieben wird, hat man k = 32 und m = 8 oder 16.

Mit dem in Figur 1 dargestellten Schaltkreis setzt man ein Verfahren um, das die folgenden Schritte aufweist:

1. Berechnen eines Parameters H ($H = 2^{2 \cdot n} \bmod N$) und eines Parameters J_0, codiert in k Bits, wobei $J_0 = -N_0^{-1} \bmod 2^k$, wobei N_0 das Wort mit dem geringsten Gewicht von Modulo N ist, und Speichern von J_0 in einem Register 17 mit k Bits,

2. Laden des Multiplikators B und von Modulo N in jeweilige Register 10 und 12 mit n Bits, wobei $n = m \cdot k$, und Initialisieren eines Registers 11 mit n Bits auf Null, wobei der Inhalt dieses Registers S genannt wird und S eine variable binäre Dateneinheit ist, die in n Bits codiert ist.

3. Durchführen einer Schleife, die mit i indiziert ist, wobei i zwischen 1 und m liegt, wobei jede i-te Iteration die folgenden Operationen aufweist:

a) Übertragen des i-ten Wortes A_{i-1} des Multiplikanden A von einem Register 16 zu einem Speicher-Flip-Flop 21,

b) Erzeugen eines Wertes $X(i) = S(i-1) + B \cdot A_{i-1}$ mit $S(0) = 0$ und $S(i-1)$ als aktualisiert genannter Wert von S, der nachfolgend definiert wird durch:

I - Verschieben nach rechts des Inhaltes des Registers 10 zum Eingang eines ersten seriellen-parallelen Multiplikatorschaltkreises 19 mit Rückschleifen des Ausgangs des Registers 10 an seinen Eingang,

II - Multiplizieren der Bits von B mit A_{i-1},

III - Verschieben nach rechts des Inhalts des Registers 12 mit Rückschleifen des Ausgangs an den Eingang,

IV - Bestimmen des aktualisierten Wertes von $S(i-1)$ als abgespeicherter Wert in dem Register 11 nach $(i-1)$-ten Iteration, wenn dieser kleiner als N ist, und wenn R größer in dem seriell N von diesem in einem ersten seriellen Subtraktionsschaltkreis 28 abgezogen wird, wobei der dabei sich ergebende Wert als aktualisierter Wert von $S(i-1)$ bezeichnet wird, und

V - Verschieben nach rechts des Inhalts des Registers 11 und Bit-für-Bit-Addieren des Wertes der Multiplikation $B \cdot A_{i-1}$ mit dem aktualisierten Wert von $S(i-1)$ in einem ersten seriellen Additionsschaltkreis 30,

c) Multiplizieren des Wortes mit dem geringsten Gewicht von $X(i)$, $X_0(i)$, mit J_0 in einem zweiten seriellen-parallelen Multiplikatorschaltkreis 20 und Eingeben des Wertes $X_0(i) \cdot J_0 \bmod 2^k = Y_0(i)$ in ein Register 18 und gleichzeitig Verzögern von N und $X(i)$ um k Zyklen in Verzögerungsschaltkreisen 32 und 34,

d) Berechnen eines Wertes $Z(i) = X(i) + Y_0(i) \cdot N$ durch:

I - Multiplizieren von $Y_0(i)$ mit N, verzögert um k Zyklen, in dem zweiten Multiplikatorschaltkreis 20, und

II - Addieren von X(i) zu dem Wert $Y_0(i) \cdot N$ in einem zweiten seriellen Additionsschaltkreis 31,

e) Verwerfen des Wortes mit dem geringsten Gewicht von Z(i) und Abspeichern von den restlichen Worten, d.h. $Z(i)/2^k$ in dem Register 11,

f) Bit-für-Bit-Vergleichen von $Z(i)/2^k$ N mit dem Ziel, den aktualisierten Wert S(i) in der folgenden Iteration zu bestimmen, wie es oben beschrieben wurde, wobei dieser Vergleich durch Bit-für-Bit-Subtraktion von $Z(i)/2^k$ und N in einem zweiten seriellen Subtraktionsschaltkreis 29 erfolgt, wobei N um zusätzliche k Zyklen verzögert wurde,

g) wobei das i-te Wort des Multiplikanden A in das Register 16 einem beliebigen Zeitpunkt während der obigen Operationen geladen wird,

4. bei der m-ten Iteration Verwerfen des Wortes mit dem geringsten Gewicht von Z(m) und Eintragen der restlichen Worte, d.h. $Z(m)/2^k$ in das Register 10,

5. Wiederholen der Schritte 3 und 4, bei denen man $Z(m)/2^k$ anstelle von B und H anstelle von A verwendet und $Z(m)/2^k$ oder $(Z(m)/2^k) - N$ über einen dritten seriellen Subtraktionsschaltkreis 27 an den Multiplikatorschaltkreis 19 weitergibt (wenn $Z(m)/2^k) \geq N$),

6. Ausgabe des in dem Register 10 gespeicherten Ergebnisses bei der letzten Iteration, unter Umständen nach Subtraktion von N, falls notwendig.

Allgemein kann man unter Bezeichnung der Zyklusdauer oder des Zyklus als Periode des Taktsignals, das die Funktion des Schaltkreises nach Figur 1 synchronisiert, die notwendige Zeit für die Bearbeitung einer Modulo-Multiplikation grundsätzlich aufteilen in:

- $n \cdot (n + 1)$ Zyklusdauern für die Berechnung von H,

- n Zyklusdauern für Schritt 2,

- $m \cdot (n + 2 \cdot k + x)$ Zykluszeiten für die Schritte 3 und 4, die zusammengefaßt wurden, einerseits und für den Schritt 5 andererseits, wobei x eine ganze Zahl ist,

- n Zyklusdauern für Schritt 6.

In der Praxis ist x eine Funktion der Initialisierung des Schaltkreises, d.h. grundsätzlich der Einrichtung von Steuersignalen (für die Multiplexer z.B.), um eine kohärente Funktion des Schaltkreises sicherzustellen. In der Praxis kann man z.B. davon ausgehen, daß $x = 7$.

Was den ersten Schritt betrifft, wird die Berechnung des Parameters J_0 durch die Zentraleinheit vorgenommen (Software-Lösung).

Man kann zeigen, daß H von der Größe des Registers 16 und der Anzahl, bei denen dieses Register in einer Schleife verwendet wird, abhängt. Man hat $H = 2^{2 \cdot n} \bmod N$. Dieser Parameter ist ein Fehlerkorrekturparameter. Tatsächlich erzeugt der Schritt 4 ein Ergebnis der Form $A \cdot B \cdot I$ mit $I = 2^{-n} \bmod N$. Man hat $H \cdot I^2 = 1 \bmod N$, was die Ausgabe eines exakten Ergebnisses erlaubt, d.h. gleich dem Ergebnis der modularen Multiplikation $A \cdot B \bmod N$ bei dem Schritt 5

des modularen Modifikationsverfahrens, das oben beschrieben wurde.

Im übrigen wird die Erzeugung von H mit Hilfe des Coprozessors nach dem folgenden Verfahren durchgeführt, das anhand von Figur 2 erläutert wird, wobei die Figur 2 der Figur 9 in der genannten europäischen Patentanmeldung entspricht.

Um H zu erzeugen geht man wie folgt vor (siehe auch die Beschreibung auf Seite 20, Zeile 47 bis Seite 25, Zeile 52 der genannten europäischen Patentanmeldung):

1. Laden von N in das Register 12 und Initialisieren des Registers 10 auf $B(0) = 0$,

2. Gleichzeitig:

- Verschieben nach rechts und Bit-für-Bit-Subtrahieren von $B(0)$ und N in einem seriellen Subtrahierer 27 mit Verschieben um eine Einheit des Ergebnisses $R(0) = B(0) - N \bmod 2^n$, wobei das Verschieben in dem Subtahierer unter Ausgabe eines ersten Bits auf 0 erfolgt,

- Laden von $B(1) = 2 \cdot R(0)$ in das Register 10,

- Bit-für-Bit-Subtrahieren von $2 \cdot R(0)$ und N, um festzustellen, ob $2 \cdot R(0) \geq$ oder $< N$, wobei diese Subtraktion in einem zweiten Subtrahierer 40 mit einem Test in einem Schaltkreis 44 des Ergebnisses des Subtraktion erfolgt,

3. Durchführen einer Schleife, die mit einem Index i versehen ist, wobei i zwischen 1 und n liegt und jede Iteration die folgenden Operationen umfaßt:

- wenn $B(i) < N$, Laden von $B(i+1) = 2 \cdot (B(i) - 0)$ in das Register 10,

- andernfalls Laden von $B(i+1) = 2 \cdot (B(i) - N)$ in das Register 10.

Man hat $B(n+1) = H = 2^{2 \cdot n} \bmod N$.

Der Erfinder hat versucht, das Durchführen der modularen Multiplikation durch den in Figur 1 dargestellten Coprozessor zu verbessern, um einerseits die notwendige Zeit für den Ablauf in bezug auf identische Größen von Dateneinheiten zu reduzieren und andererseits die Möglichkeiten der Berechnung durch den Schaltkreis zu erweitern.

Damit wird als erfindungsgemäßes Verfahren zur Durchführung der Modulo-Multiplikation nach dem Verfahren von Montgomery vorgeschlagen, bei welchem ein Multiplikant A und ein Multiplikator B jeweils mit a und b Worten à k Bits codiert werden, wobei die Worte mit dem höchsten Gewicht von A und B nicht 0 sind, ein Modulo N codiert ist in m Worte à k Bits, wobei das Modulo (m - m') Worte mit dem höchsten Gewicht à k Bits hat, die 0 sind, wobei $0 < m' \leq m$, das die Schritte der Multiplikation in einem Multiplikatorschaltkreis mit einem seriellen Eingang zum Empfangen von Daten, die in wenigstens m' Worten à k Bits codiert sind, einem parallelen Eingang zum Empfangen von Worten, die k Bits codiert sind, und einem seriellen Ausgang, umfaßt, dadurch gekennzeichnet, daß man sukzessive beim Durchführen der Multiplikation eine vorgegebene Zahl p von Worten am parallelen Eingang des Multiplikatorschaltkreises anlegt, wobei p unabhängig von m ist und größer oder gleich der Zahl a ist.

In dem existierenden Schaltkreis verwendet man eine Codierung von Daten (Multiplikant, Multiplikator, Modulo) vom eingefrorenen Typ, d.h. unabhängig von Werten von diesen. Tatsächlich betrachte man die codierten Daten einer festen Größe mit m · k Bits, was zur Durchführung von überflüssigen Operationen führt, wenn die Operanden Worte mit größerem Gewicht à k Bits gleich Null umfassen, insbesondere wenn dies den Multiplikanden betrifft. Die Erfindung erlaubt es, eine Reduktion der Zahl an Operationen ins Auge zu fassen, und insbesondere die Zahl der Verwendungen des Registers 16. Sie erlaubt es im übrigen, eine modulare Multiplikation durchzuführen, was immer die Größe des Multiplikanden in Anzahl Bits sei.

Erfindungsgemäß wird ein Verfahren vorgeschlagen, das vorzugsweise gekennzeichnet ist:

- Erzeugen von $H = 2^{(a+b)\cdot k} \bmod N$,

- Erzeugen eines Zwischendatensatzes, der codiert ist in m Worten à k Bits, indem die m Worte von H und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden, und

- Erzeugen von $A \cdot B \bmod N$, indem die m Worte des Zwischendatensatzes und die b Worte von B jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden.

So kann man die Operationen auf den Operanden, Multiplikand und Multiplikator, beliebiger Größe durchführen. Insbesondere können die Operanden beide in einer Zahl von Worten größer als m codiert werden und größer sein als das Modul.

Erfindungsgemäß wird außerdem vorgeschlagen, wenn B kleiner oder gleich groß N ist, daß:

- $H = 2^{(a+m')\cdot k} \bmod N$ erzeugt wird,

- ein Zwischendatensatz erzeugt wird, der in m Worte à k Bit codiert ist, indem m Worte, die B in m Worten à k Bit entsprechen, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden, und

- $A \cdot B \bmod N$ erzeugt wird, indem die m Worte des Zwischendatensatzes und die m' Worte mit dem geringsten Gewicht von H jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden.

Erfindungsgemäß wird außerdem ein Verfahren vorgeschlagen, das dadurch gekennzeichnet ist, daß man (a + b) und m' vergleicht und

- wenn $a + b < m'$, dann $A \cdot B \bmod N$ erzeugt wird, indem m Worte entsprechend $B \cdot 2^{a\cdot k}$ codiert in m Worte ausgegeben und die a Worte von A jeweils am seriellen Eingang und parallelen Eingang des Multiplikatorschaltkreises angelegt werden,

- wenn $a + b = m'$, dann $B \cdot 2^{2\cdot k}$ und N verglichen werden, und

- wenn $B \cdot 2^{a\cdot k} < N$, dann $A \cdot B \bmod N$ erzeugt wird, indem m Worte, entsprechend $B \cdot 2^{a-k}$ codiert in m Worten und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden,

- andernfalls $A \cdot B \bmod N$ erzeugt wird, indem m Worte entsprechend $B \cdot 2^{2 \cdot k} \bmod N$ codiert in m Worten und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden.

So kann man in bestimmten Fällen eine modulare Multiplikation durchführen, ohne den Fehlerkorrekturparameter H berechnen zu müssen.

Die Erfindung ergibt sich neben weiteren Einzelheiten und Vorteilen aus der folgenden Beschreibung von Ausführungsformen der Erfindung, die als Beispiel ohne Einschränkung dargestellt werden, wobei Bezug genommen wird auf die beigefügten Zeichnungen, bei denen:

- Figur 1 schematisch einen Schaltkreis zur Durchführung einer modularen Multiplikation nach dem Verfahren von Montgomery zeigt,

- Figur 2 einen Schaltkreis zu der Erzeugung eines Fehlerkorrekturparameters gemäß der Erfindung zeigt.

Figur 1 zeigt einen Schaltkreis 1 zur Verarbeitung von modularen Operationen.

Er umfaßt:

- drei Schieberegister 10, 11 und 12 mit seriellem Eingang und Ausgang. Diese Register umfassen jeweils die gleiche Zahl n von Zellen mit $n = m \cdot k$. Diese Register können unterteilbar sein, z.B. in Register mit n/2 Zellen und in Register von k Bits bei den Registern 10 und 12.

- Multiplexern 13, 14 und 15, die jeweils vor den Registern 10, 11 und 12 angeordnet sind. Man ordnet ebenso

Multiplexer vor die Unterteilungen an, wenn diese dann existieren.

- drei Register 16, 17 und 18, die jeweils k Zellen umfassen. Die Register 16 und 17 und 18 sind Register mit parallelem Ausgang und seriellem Eingang.

- zwei Multiplikationsschaltkreisen 19 und 20, die jeweils einen seriellen Eingang, einen parallelen Eingang und einen seriellen Ausgang haben. Der parallele Eingang des Multiplikationsschaltkreises 19 ist mit dem Ausgang des Registers 16 über die Speicherkippstufe 21 mit k Zellen verbunden. Der parallele Eingang des Multiplikationsschaltkreises 20 ist mit einem der Ausgänge der Register 17 oder 18 über eine Speicherkippstufe 22 mit k Zellen verbunden. Diese Kippstufe 22 ist selbst verbunden mit einem der Ausgänge der Register 17 und 18 über einen Multiplexer mit 2 parallelen Eingängen und einem parallelen Ausgang.

- Multiplexern 24, 25, 25', 26, 36 und 38.

- einen Demultiplexer 39.

- serielle Subtraktionsschaltkreise 27, 28 und 29.

- serielle Additionsschaltkreise 30 und 31.

- Verzögerungsschaltkreise 32, 33 und 34 zum Verzögern um k Zyklen der Fortpflanzung von binären Daten.

- einen Schaltkreis 35 zum Abspeichern des Vergleichsergebnisses.

Weitere Einzelheiten findet man in der genannten europäischen Patentanmeldung (EP - 0 601 907 A2) und ins-

besondere in Figur 3 dieser Anmeldung und bei den Abschnitten der Beschreibung, die sich darauf beziehen: Seite 15, Zeile 54 bis Seite 16, Zeile 13 und Seite 17, Zeile 50 bis Seite 18, Zeile 55.

Der Schaltkreis nach Figur 1 ermöglicht die Umsetzung der Erfindung.

Im folgenden der Beschreibung geht man davon aus, daß:

- ein Multiplikant A (binäre Daten) und ein Multiplikator B (binäre Daten) codiert sind in jeweils a bzw. b Worte à k Bits, wobei die Worte mit dem größten Gewicht von A und B nicht Null sind,

- ein Modul N (binäre Daten) codiert ist in m Worten à k Bits, wobei das Modul (m - m')Worte mit dem größten Gewicht à k Bits mit dem Wert Null haben, wobei $0 < m' \leq m$.

Unter genutzter Größe binärer Daten versteht man die minimale Anzahl von Worten à k Bits, die notwendig und ausreichend ist, um Daten darzustellen, d.h. die minimale Zahl von Worten, so daß das Wort mit dem höchsten Gewicht wenigstens ein Bit gleich Eins enthält. Also: A und B und N haben eine genutzte Größe, die jeweils a, b und m' ist.

Unter Zahl von genutzten Bits versteht man die minimale Zahl von notwendigen und ausreichenden Bits, um Daten darzustellen, d.h. die minimale Zahl von Bits, so daß das Bit mit dem höchsten Gewicht gleich 1 ist.

Figur 2 zeigt einen Schaltkreis, der umfaßt:

- die zwei Schieberegister 10 und 12, den Subtraktionsschaltkreis 27, den Multiplexer 36,

- zwei Multiplexer mit zwei Eingängen 41 und 42, bei denen die jeweiligen Ausgänge jeweils mit den Eingängen der Register 10 und 12 verbunden sind,

- einen seriellen Subtraktionsschaltkreis 40,

- ein NICHT-UND-Gatter 43 mit zwei Eingängen,

- einen Schaltkreis 44 zum Abspeichern des Vergleichsergebnisses.

Die Subtraktionsschaltkreise 27 und 40 haben zwei serielle Eingänge und einen seriellen Ausgang.

Der Subtraktionsschaltkreis 27 hat einen ersten Eingang, der mit dem Ausgang des Registers 10 verbunden ist, und sein Ausgang ist mit einem ersten Eingang des Subtraktionsschaltkreises 40 verbunden. Der Subtraktionsschaltkreis 40 ist über seinen zweiten Eingang mit dem Ausgang des Registers 12 verbunden, und sein Ausgang ist mit einem invertierenden Eingang des Gatters 43 verbunden.

Der andere (nicht invertierende) Eingang des Gatters 43 ist mit dem Ausgang des Subtraktionsschaltkreises 27 verbunden. Sein Ausgang ist mit einem Eingang des Schaltkreises 44 verbunden. Dieser Schaltkreis 44 hat einen anderen Eingang, um ein Reinitialisierungssignal RES zu empfangen.

Der Multiplexer 36 hat zwei Eingänge und einen Ausgang. Sein Ausgang ist mit dem zweiten Eingang des Subtraktionsschaltkreises 28 verbunden. Seine Eingänge sind jeweils mit dem Ausgang des Registers 12 und Masse (Potential, das logisch 0 entspricht) verbunden. Der Multiplexer 36 verbindet selektiv seinen Ausgang mit seinem

ersten oder seinem zweiten Eingang, je nach dem Zustand eines Auswahlsignals SC, das er von dem Schaltkreis 44 erhält (z.B. mit dem ersten Eingang, wenn SC = 0, und mit seinem zweiten Eingang, wenn SC = 1).

Der Multiplexer 41 verbindet seine Eingänge jeweils mit dem Ausgang des Subtraktionsschaltkreises 27 und Masse.

Der Multiplexer 42 verbindet seine Eingänge jeweils mit dem Ausgang des Registers 12 und einem Eingangsanschluß, um seriell binäre Daten zu empfangen (in der Praxis das Modul N).

Der Schaltkreis nach Figur 2 wird verwendet, um einen Fehlerkorrekturparameter H zu erzeugen, der eine binäre Dateneinheit ist.

Man möchte eine binäre Dateneinheit erzeugen, die A · B mod N entspricht.

Mehrere Verfahren sind denkbar.

- Wenn a + b > m':

werden folgende Schritte durchgeführt:

P1 - Erzeugen eines Fehlerkorrekturparameters $H = 2^{(a+b)\cdot k} \bmod N$ mit einer genutzten Größe, die kleiner oder gleich m' ist, und eines Parameters J_0, codiert in k Bits mit $J_0 = N_0^{-1} \bmod 2^k$, wobei N_0 das Wort mit dem geringsten Gewicht von Modulo N ist, und Abspeichern von J_0 in dem Register 17 à k Bits,

P2 - Laden des Parameters H in dem Register 10 und von Modulo N in dem Register 12 und Initialisieren des

Registers 11 mit $n = m \cdot k$ Bits auf Null, wobei der Inhalt dieses Registers 11 S genannt wird und S eine variable binäre Dateneinheit ist, die in n Bits codiert ist,

P3 - Durchführen einer Schleife, die mit i indiziert ist, wobei i zwischen 1 und a liegt, wobei jede i-te Iteration die folgenden Operationen beinhaltet:

a) Übertragung von dem i-ten Wort A_{i-1} des Multiplikanten A des Registers 16 zur Speicherkippstufe 21,

b) Erzeugen eines Wertes $X(i) = S(i-1) + H \cdot A_{i-1}$, wobei $S(0) = 0$ (n Bits auf Null) und $S(i-1)$ der aktualisierte Wert von S ist, der im folgenden definiert ist durch:

I - Verschieben nach rechts des Inhalts des ersten Registers 10 zum Eingang des seriell-parallelen Multiplikatorschaltkreises 19 mit Rückschleifen des Ausgangs des Registers 10 an seinen Eingang,

II - Multiplizieren der Bits von H mit A_{i-1},

III - Verschieben nach rechts des Inhalts des Registers 12 mit Rückschleifung des Ausgangs des Registers an seinen Eingang,

IV - Bestimmen des aktualisierten Wertes von $S(i-1)$ als Wert, der in dem Register 11 nach der $(i-1)$-ten Iteration abgelegt wurde, wenn dieser kleiner als N ist, und durch serielles Subtrahieren von N von diesem, wenn der Wert größer ist, wobei der Ergebniswert der aktualisierte Wert von $S(i-1)$ ist, und

V - Verschieben nach rechts des Inhalts des Registers 11 und Addieren des Wertes der Multiplikation $H \cdot A_{i-1}$ Bit für Bit zu dem aktualisierten Wert von S(i-1),

c) Multiplikation des Wortes mit dem geringsten Gewicht von X(i), $X_0(i)$, mit J_0 und Eintragen des Wertes $X_0(i) \cdot J_0 \bmod 2^k = Y_0(i)$ in das Register 18 und gleichzeitiges Verzögern von N und X(i) um k Zyklen, zu den Verzögerungsschaltkreisen 32 und 34,

d) Berechnen eines Wertes $Z(i) = X(i) + Y_0(i) \cdot N$ durch:

I - Multiplizieren von $Y_0(i)$ mit N, verzögert um k Zyklen, zu dem Multiplikationsschaltkreis 20 und

II - Addieren von X(i) zu dem Wert $Y_0(i) \cdot N$ in dem Additionsschaltkreis 31

e) Vernachlässigen des Wortes mit dem geringsten Gewicht von Z(i) und Abspeichern der übrigen Worte, d.h. $Z(i)/2^k$, in dem Register 11,

f) Bit-für-Bit-Vergleichen von $Z(i)/2^k$ mit N, um den aktualisierten Wert S(i) der folgenden Iteration auf die oben beschriebene Art zu bestimmen, wobei dieser Vergleich erfolgt durch Bit-für-Bit-Subtrahieren von $Z(i)/2^k$ und N in dem Subtraktionsschaltkreis 29 erfolgt, wobei N um zusätzliche k Zyklen in dem Verzögerungsschaltkreis 33 verzögert wurde,

g) wobei das i-te Wort des Multiplikanden A in dem Register 16 zu irgendeinem Zeitpunkt während der obengenannten Operation abgelegt worden ist;

P4 - bei der a-ten Iteration Ignorieren des Wortes mit dem geringsten Gewicht von Z(a) und Eintragen der restlichen Worte, d.h. $Z(a)/2^k = A \cdot 2^{b \cdot k} \bmod N$, in dem Register 10,

P5 - Wiederholen der Schritte P3 und P4 bei welchen:

- A durch B ersetzt wird und $Z(m)/2^k$ oder $(Z(m)/2^k) - N$ durch den Subtraktionsschaltkreis 27 an den Multiplikatorschaltkreis 19 ausgegeben wird,

- i zwischen 1 und b variiert wird,

P6 - Ausgeben des in dem Register 10 gespeicherten Ergebnisses für die letzte Iteration, eventuell nach Abzug von N durch den Subtraktionsschaltkreis 27, falls nötig.

In dem obigen Verfahren verwendet man p mal das Register 16, wobei $p = a + b$.

Man stellt fest, daß man $a + b > 2 \cdot m$ wählen kann und $b > m'$, was nicht möglich war bei dem früheren Berechnungsmodus.

Die Zeit für die notwendige Berechnung zum Bestimmen des erwünschten Ergebnisses wird $(a + b) \cdot (n + 2 \cdot k + x)$ Zyklen betragen anstelle von $2 \cdot m \cdot (n + 2 \cdot k + x)$ Zyklen bei dem Stand der Technik.

Unter Zyklen versteht man die Periode des Taktsignals, das die Funktion des Schaltkreises nach Figur 1 synchronisiert.

In der Praxis ist x eine Funktion der Initialisierung des Schaltkreises, d.h. grundsätzlich der Einrichtung von Steuersignalen (für Multiplexer z.B.), um eine kohärente Funktion des Schaltkreises zu gewährleisten.

Was die Erzeugung des Fehlerkorrekturparameters H betrifft, geht man nach den folgenden Schritten vor, die sich auf die Figur 2 beziehen:

H1 - Ablegen von Modulo N in dem Register 12, Initialisieren des Registers 10, B(0) genannt, und Initialisieren des Registers 44 (d.h. Erzeugen eines Signals RES, so daß SC = 0),

H2 - Ausgeben von N in Register 12 durch Verschieben nach rechts mit Rückschleifung an seinen Eingang von l Bits des ersten Registers, wobei l die Zahl der genutzten Bits von Modulo N ist, um ein Bit mit höchstem Gewicht auf 1 in dem Register 12 zu haben,

H3 - Erzeugung und Abspeichern eines Datenwertes $B(1) = 2 \cdot (B(0) - N')$, codiert in n Bit mit $N' = N \cdot 2^{n-1}$, wobei N' dem im Register 12 gespeicherten binären Datenwert entspricht, durch:

- Ausgeben von N' und B(0) durch Verschieben nach rechts in den Registern 10 und 12 der Inhalte der Register, wobei der Eingang des Registers 10 mit dem Ausgang des Subtraktionsschaltkreises 27 verbunden ist und der Eingang des Registers 12 mit seinem Ausgang verbunden ist,

- Bit-für-Bit-Subtraktion in dem Substraktionsschaltkreis 27 je nach ihrem Ausgang von Bits von N' und B(0) mit Verschieben nach links des $R(0) = B(0) - N'$ genannten Ergebnisses um eine Einheit,

- Ablegen des Resultats der Subtraktion nach Verschieben, genannt $B(1) = 2 \cdot R(0)$, in das Register 10,

- Bit-für-Bit-Subtraktion in dem Subtraktionsschaltkreis 40 von B(1) und N', um zu bestimmen, ob B(1) ≥ oder < N', und Erzeugen im Schaltkreis 44 von SC = 0 für B(1) ≥ N' und von SC = 1 für B(1) < N',

H4 - Erzeugen eines Datenwertes $H_{int} = 2^v \bmod N'$ mit $v = n - l + m' \cdot k + (a + b - m') \cdot k/2^r$ mit r als ganzer Zahl, so daß auch $k/2^r$ eine ganze Zahl ist, durch:

Durchführen einer mit i indizierten Schleife, wobei i eine ganze Zahl zwischen 1 und v ist, wobei jede i-te Iteration die folgenden Operationen umfaßt:

- wenn B(i) < N', dann Ablegen von $B(i+1) = 2 \cdot (B(i)-0) = 2 \cdot B(i)$ nach Verschieben nach links von B(i) um eine Einheit und Bit-für-Bit-Vergleich von B(i+1) und N',

- andernfalls Bit-für-Bit-Subtraktion von N' und B(i) mit Verschieben nach links um eine Einheit des Ergebnisses und Ablegen in dem Register 10 von $B(i+1) = 2 \cdot (B(i) - N')$ und Bit-für-Bit-Vergleich von B(i+1) und N',

H5 - wenn B(v+1) ≥ N': Bit-für-Bit-Subtraktion von B(v+1) und N' in dem Subtraktionsschaltkreis 27, was $2 \cdot (B(v+1) - N')$ ergibt, und Ablegen von B(v+1) - N' in dem Register 10, was ein Verschieben um eine Einheit nach rechts erfordert,

H6 - Verschieben nach rechts um n - 1 Bits in dem Register 10 und 12,

H7 - Erzeugen des Parameters H durch Durchführen von r P_{field}-Operationen:

$H_{int}(j) = P(H_{int}(j-1), H_{int}(j-1))_N$ mit j Index zwischen 1 und r, und $H_{int}(0) = B(v+1) \cdot 2^{l-n}$ oder $(B(v+1) - N') \cdot 2^{l-n}$.

$H_{int}(0) = 2^{v'} \bmod N$, mit $v' = m' \cdot k + (a + b - m') : k/2^r$.

Man erhält so $H = 2^{(a+b)\cdot k} \bmod N$.

Zur Durchführung der Operationen P_{field} geht man nach den folgenden Schritten vor:

Es sei j = 1,

1. - Erzeugen eines Parameters J_0, codiert in k Bits mit $J_0 = - N_0^{-1} \bmod 2^k$, wobei N_0 das Wort mit dem geringsten Gewicht von Modulo N ist, und Abspeichern von J_0 in dem Register 17 à k Bits.

2. - Initialisieren des Registers 11 mit n = m · k Bits auf Null, wobei der Inhalt dieses Registers 11 mit S bezeichnet wird, wobei S eine variable binäre Dateneinheit ist, die in n Bits codiert ist.

3. - Durchführen einer Schleife, die mit i indiziert ist, wobei i zwischen 1 und m' liegt, wobei jede i-te Iteration die folgenden Operationen umfaßt:

a) Übertragung von dem i-ten Wort H_{i-1} des Multiplikanten $H_{int}(j-1)$ des Registers 16 zur Speicherkippstufe 21,

b) Erzeugen eines Wertes $X(i) = S(i-1) + H \cdot H_{i-1}$, wobei $S(0) = 0$ (n Bits auf Null) und S(i-1) der aktualisierte Wert von S ist,

c) Multiplikation des Wortes mit dem geringsten Gewicht von X(i), $X_0(i)$, mit J_0 zu dem Multiplikationsschaltkreis 20 und Eintragen des Wertes $X_0(i) \cdot J_0 \bmod 2^k = Y_0(i)$ in das Register 18 und gleichzeitiges Verzögern von N und X(i) um k Zyklen in den Verzögerungsschaltkreisen 32 und 34,

d) Berechnen eines Wertes $Z(i) = X(i) + Y_0(i) \cdot N$ durch:

I - Multiplizieren von $Y_0(i)$ mit N, verzögert um k Zyklen, in dem Multiplikatorschaltkreis, und

II - Addieren von X(i) zu dem Wert $Y_0(i) \cdot N$ in dem Additionsschaltkreis 31,

e) Vernachlässigen des Wortes mit dem geringsten Gewicht von Z(i) und Abspeichern der übrigen Worte, d.h. $Z(i)/2^k$ in dem Register 11,

f) Bit-für-Bit-Vergleichen von $Z(i)/2^k$ mit N, um den aktualisierten Wert S(i) der folgenden Iteration auf die oben beschriebene Art zu bestimmen, wobei der Vergleich durch Bit-für-Bit-Subtrahieren von $Z(i)/2^k$ und N in dem Subtraktionsschaltkreis 29 erfolgt, wobei N um zusätzliche k Zyklen in dem Verzögerungsschaltkreis 33 verzögert wurde,

g) wobei das i-te Wort des Multiplikanden $H_{int}(j-1)$ in dem Register 16 zu irgendeinem Zeitpunkt während der obengenannten Operation abgelegt worden ist;

4. - bei der m-ten Iteration Ignorieren des Wortes mit dem geringsten Gewicht von Z(m') und Eintragen der restlichen Worte, d.h. $Z(m')/2^k$ in dem Register 10,

5. - Wiederholen der Schritte 3 und 4 (r-1)-mal, wobei $H_{int}(j-1)$ durch $H_{int}(j)$ ersetzt wird,

6. - Ausgeben des in dem Register 10 gespeicherten Ergebnisses für die letzte Iteration, eventuell nach Abzug von N im Subtraktionsschaltkreis 27, falls nötig.

Was das Erzeugen des Parameters J_0 betrifft, so kann dies vorher erfolgen beim Umsetzen von Modulo-Operationen. Tatsächlich wird es durch die Zentraleinheit durchgeführt und ist daher unabhängig von Coprozessor.

Man stellt fest, daß erfindungsgemäß ein Fehlerkorrekturparameter H berechnet wird, indem einerseits Subtraktionen kleinerer Zahlen als bei dem bekannten Verfahren und andererseits Operationen P_{field} vorgeschlagen werden, was es ermöglicht, eventuell die notwendige Zeit zum Realisieren des Schrittes P1 des Multiplikationsschaltkreises zu verringern.

Im Stand der Technik erfordert die Berechnung von H $n \cdot (n + 1)$ Zyklen.

Nach dem obigen Verfahren ist die notwendige Zeit $n \cdot (m' \cdot k - 1 + (a + b - m') \cdot k/2^r + 1) + r \cdot m' \cdot (n + 2 \cdot k + x)$.

- Wenn $a + b > m'$ und $B < N$:

Wenn B kleiner als N, kann man das obige Verfahren modifizieren, indem:

- $H = 2^{(a+m')\cdot k} \bmod N$ erzeugt wird,

- ein Zwischendatensatz $C = P(A, B)_N = A \cdot B \cdot I_a \bmod N$, codiert in n Bits, erzeugt wird, wobei $I_a = 2^{-a\cdot k} \bmod N$ Fehler von A, indem m Worte, die B in m Worten à k Bit

entsprechen, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden,

- $P(H,C)_N = H \cdot C \cdot I_h \bmod N$, erzeugt wird, wobei $I_h = 2^{-m \cdot k} \bmod N$, indem die m Worte des Zwischendatensatzes und die m' Worte mit dem geringsten Gewicht von H jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden.

Das Verfahren zum Durchführen der modularen Multiplikation umfaßt daher die folgenden Schritte:

P1 - Erzeugung des Fehlerkorrekturparameters H und eines Parameters J_0, der codiert ist in k Bit, und Abspeichern von J_0 in dem Register 17,

P2 - Ablegen des Parameters B und von Modulo N in den jeweiligen und Registern 10 und 12 und Initialisieren des Registers 11 auf Null, wobei der Inhalt dieses dritten Registers mit S bezeichnet wird,

P3 - Durchführen einer Schleife, die mit i indiziert ist, wobei i eine Variable zwischen 1 und a ist, wobei jede i-te Iteration die folgenden Operationen umfaßt:

a) Übertragung von dem i-ten Wort A_{i-1} des Multiplikanten A des Registers 16 an die Speicherkippstufe 21,

b) Erzeugen eines Wertes $X(i) = S(i-1) + B \cdot A_{i-1}$, wobei $S(0) = 0$ und $S(i-1)$ gleich dem aktualisiert genannten Wert S ist, der wie nachfolgend definiert ist:

I - Verschieben nach rechts des Inhalts des Registers 10 zum Eingang des seriell-parallelen Multiplikatorschalt-

kreises 19 mit Rückschleifen des Ausgangs des Registers 10 an seinen Eingang,

II - Multiplizieren der Bits von B mit A_{i-1},

III - Verschieben nach rechts des Inhalts des Registers 12 mit Rückschleifung des Ausgangs des Registers an seinen Eingang,

IV - Bestimmen des aktualisierten Wertes von S(i-1) als Wert, der in dem Register 11 nach der (i-1)-ten Iteration abgelegt wurde, wenn dieser kleiner als N ist, und nach seriellem Subtrahieren von N von diesem, wenn der Wert größer ist, wobei der Ergebniswert der aktualisierte Wert von S(i-1) ist, und

V - Verschieben nach rechts des Inhalts des Registers 11 und Bit-für-Bit-Addieren des Wertes der Multiplikation $B \cdot A_{i-1}$ zu dem aktualisierten Wert von S(i-1),

c) Multiplikation des Wortes mit dem geringsten Gewicht von X(i), $X_0(i)$, mit J_0 und Eintragen des Wertes $X_0(i) \cdot J_0 \bmod 2^k = Y_0(i)$ in ein Register 18 und gleichzeitiges Verzögern von N und X(i) um k Zyklen,

d) Berechnen eines Wertes $Z(i) = X(i) + Y_0(i) \cdot N$ durch:

I - Multiplizieren von $Y_0(i)$ mit N, verzögert um k Zyklen, und

II - Addieren von X(i) zu dem Wert $Y_0(i) \cdot N$,

e) Vernachlässigen des Wortes mit dem geringsten Gewicht von Z(i) und Abspeichern der übrigen Worte, d.h. $Z(i)/2^k$ in dem Register 11,

f) Bit-für-Bit-Vergleichen von $Z(i)/2^k$ mit N, um den aktualisierten Wert S(i) der folgenden Iteration auf die oben beschriebene Art zu bestimmen,

g) wobei das i-te Wort des Multiplikanden A in dem Register 16 zu irgendeinem Zeitpunkt während der obengenannten Operation abgelegt worden ist;

P4 - bei der a-ten Iteration Ignorieren des Wortes mit dem geringsten Gewicht von Z(a) und Eintragen der restlichen Worte in dem Register 10,

P5 - Wiederholen der Schritte P3 und P4 bei welchen:

- A durch H ersetzt wird und $Z(a)/2^k$ oder $(Z(a)/2^k) - N$ an den ersten Multiplikatorschaltkreis 19 ausgegeben wird,

- i zwischen 1 und b variiert wird;

P6 - Ausgeben des in dem Register 10 gespeicherten Ergebnisses für die letzte Iteration, eventuell nach Abzug von N, falls nötig.

Man kann A in einer Zahl a Worte größer als m codieren.

Um H zu berechnen, geht man wie folgt vor (siehe Figur 2):

H1 - Ablegen von Modulo N in dem Register 12 und Initialisieren des Registers 10 auf Null, wobei der Inhalt des Registers B(0) genannt wird,

H2 - Verschieben nach rechts mit Rückschleifung an seinen Eingang von l Bits des ersten Registers, wobei l die Zahl der für die Moduloberechnung genutzten Bits ist,

H3 - Erzeugung und Abspeichern eines Datenwertes $B(1) = 2 \cdot (B(0) - N')$, codiert in n Bit mit $N' = N \cdot 2^{n-1}$, durch:

- Verschieben nach rechts in den zwei Registern und Bit-für-Bit-Subtraktion der Inhalte der Register mit Verschieben des Resultats der Bit-für-Bit-Substraktion, genannt R(0), um eine Einheit nach links,

- Ablegen des Resultats der Subtraktion nach Verschieben, genannt B(1), in dem Register 10,

- Vergleichen von B(1) und N',

H4 - Erzeugen eines Datenwertes $H_{int} = 2^v \bmod N'$ mit $v = n - 1 + m' \cdot k + a \cdot k/2^r$ mit r als ganzer Zahl, so daß auch $k/2^r$ eine ganze Zahl ist, durch:

Durchführen einer mit i indizierten Schleife, wobei i eine ganze Zahl zwischen 1 und v ist, wobei jede i-te Iteration die folgenden Operationen umfaßt:

- wenn $B(i) < N'$, dann Ablegen von $B(i+1) = 2 \cdot B(i)$ in dem Register 10 nach Verschieben nach links von B(i) um eine Einheit und Bit-für-Bit-Vergleich von B(i+1) und N',

- andernfalls Bit-für-Bit-Subtraktion von N' und B(i) mit Verschieben nach links um eine Einheit des Ergebnisses und Ablegen in dem zweiten Register von $B(i+1) = 2 \cdot (B(i) - N')$ und Bit-für-Bit-Vergleich von B(i+1) und N',

H5 - wenn $B(v+1) \geq N'$: Bit-für-Bit-Subtraktion von B(v+1) und N' und Ablegen von $B(v+1) - N'$ in dem Register 10,

H6 - Verschieben nach rechts um n - 1 Bits in den Registern 10 und 12,

H7 - Erzeugen des Parameters H durch Durchführen von r P_{field}-Operationen:

$H_{int}(j) = P(H_{int}(j-1), H_{int}(j-1))_N$ mit j Index zwischen 1 und r, und $H_{int}(0) = B(v+1) \cdot 2^{l-n}$ oder $(B(v+1) - N') \cdot 2^{l-n}$.

Für die unterschiedlichen Berechnungen von H kann man z.B. $r = \log_2(k) - 1$ nehmen.

In bezug auf das Verfahren zum Erzeugen von H, das oben erläutert wurde, stellt man eine Modifikation des Parameters v fest, um eine Anpassung an den vorliegenden Fall zu erzielen.

- wenn $a + b \leq m'$:

kann man zwei Fälle unterscheiden:

- wenn $a + b < m'$ wird $A \cdot B \bmod N$ erzeugt, indem m Worte entsprechend $B \cdot 2^{a \cdot k}$, codiert in m Worte, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises 19 angelegt werden,

- wenn $a + b = m'$,

vergleicht man $B \cdot 2^{a \cdot k}$ und N, und

- wenn $B \cdot 2^{a \cdot k} < N$ wird $A \cdot B \bmod N$ erzeugt, indem m Worte, entsprechend $B \cdot 2^{a \cdot k}$, codiert in m Worte, und die a Worte von A jeweils am seriellen Eingang und am

parallelen Eingang des Multiplikatorschaltkreises 19 angelegt werden,

- andernfalls wird $A \cdot B \bmod N$ erzeugt, indem m Worte, entsprechend $B \cdot 2^{a \cdot k} \bmod N$, codiert in m Worte, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises 19 angelegt werden.

So wird man unter diesen Bedingungen nicht gezwungen, einen Fehlerkorrekturparameter H zu erzeugen, und man wird eine einzige Rechenoperation zum Erhalten des Ergebnisses benötigen. Die Fehlerkorrektur wird an den Operanden B weitergegeben, indem man ihn ersetzt durch $B \cdot 2^{a \cdot k} \bmod N$, was einer einfachen Verschiebung nach links von B entspricht (eventuell angepaßt durch eine Subtraktion, wenn $B \cdot 2^{2 \cdot k} \geq N$).

Man erzeugt so $P(A, B \cdot 2^{a \cdot k})_N = A \cdot B \bmod N$.

Man sieht hier den gesamten Vorteil, den es hat, den parallelen Eingang des Multiplikatorschaltkreises 19 mehrmals unabhängig von m zu benutzen.

In bezug auf den Schaltkreis, wie er im Stand der Technik existiert, ist man auf jedem Fall gezwungen, den Schaltkreis zur Sequenzierung, der die für die Funktion des in Figur 1 dargestellten Schaltkreises notwendigen Befehle erzeugt, zu modifizieren. So wird man einen programmierbaren Zähler verwenden, um einerseits die Verwendung des Registers 16, da man unterschiedliche Male dieses Register verwenden wird, und andererseits die Verschiebungen in den Registern beim Erzeugen von H in Abhängigkeit von der Anzahl der verwendeten Bits des Moduls zu verwalten.

Um die Vergleiche von a, b und m durchzuführen, wobei das schnellste Verfahren gewählt werden können soll, wird man typischerweise die Zentraleinheit verwenden, die mit dem arithmetischen Coprozessor zusammenhängt. Man kann eventuell diese Vergleiche mit Hilfe eines Schaltkreises verwirklichen, der aus Subtraktions- und Additionsschaltkreisen zusammengesetzt ist, die speziell für diesen Effekt entwickelt wurden.

Bei dem Schaltkreis, wie er beim Stand der Technik existiert, verwendet man die Register 10, 11 und 12, die konfigurierbar sind, als Register mit 8 oder 16 Worten à 32 Bits. Wenn man die Rechenzeit für eine modulare Multiplikation noch weiter verringern möchte, kann man konfigurierbare Register von 1 bis 16 Worten à 32 Bit verwenden. Dieses entspricht in der Praxis, davon auszugehen, daß m variabel ist, wie z.B., daß m = m'. Für die Durchführung reicht es, zusätzliche Multiplexer in dem Rechenschaltkreis in Höhe der Register 10, 11 und 12 hinzuzufügen. Dadurch wird es möglich, daß zu ungunsten des Umfangs die notwendige Zeit für die Verschiebungen in den Registern 10, 11 und 12 verringert wird, indem die nutzbare Größe von manipulierten binären Datensätzen angepaßt wird.

Die Verwaltung des Registers 16, wie es oben beabsichtigt ist, kann vorteilhaft ausgedehnt werden auf das Durchführen der folgenden Berechnungen.

RSA-Verfahren

Das Kryptographieverfahren RSA erfordert es, Berechnungen von Typ $C = M^D \bmod N$, wobei M eine zu verschlüsselnde oder entschlüsselnde Nachricht, N ein Modul wie $N = P \cdot Q$, mit P und Q als Primzahlen und D so gewählt ist, daß $D \cdot E = 1 \bmod ((P - 1 \cdot (Q - 1))$, bei bekanntem E zu berechnen.

Ein Algorithmus zum Durchführen dieser Berechnung ist der folgende:

$A = (M \bmod P)^D \bmod (P - 1) \bmod P$

$B = (M \bmod Q)^D \bmod (Q - 1) \bmod Q$

$U = Q^{-1} \bmod P$

Wenn $A < B \bmod P$, dann:

$C = (((A + P - (B \bmod P)) \cdot U) \bmod P) \cdot Q + B,$

andernfalls

$C = (((A - (B \bmod P)) \cdot U) \bmod P) \cdot Q + B.$

Dieser Algorithmus erfordert insbesondere:

- 2 modulare Reduktionen,
- 2 modulare Potenzierungen,
- 1 modulare Multiplikation.

Es ist insbesondere interessant, die Verfahren gemäß der Erfindung durchzuführen, um diese Berechnungen schnellstmöglich durchführen zu können.

Der Erfinder schlägt u.a. vor, die verwendeten Register zu modifizieren, z.B. indem modulierbare Register verwendet werden, die wahlweise, 32, 256, (256 + n), 384, (384 + n), 512 oder (512 + n) Zellen haben, wobei n eine ganze Zahl ist. Man nimmt z.B. n = 32.

Man wendet sich der Wahl von Modulo codiert in 512, 768 und 1024 Bits zu.

Die Wahl von mudulierbaren Registern, wie sie oben definiert wurden, erlaubt folgendes:

- in dem allgemeinen Rahmen von modularen Operationen sei Zahl von Verschiebungen in den Registern zu minimieren, indem flexibler an die verwendeten Größen von verarbeiteten Daten eine Anpassung erfolgt,

- in dem speziellen Rahmen des Umsetzens der RSA-Codierung, die Wahl unterschiedlicher Größen für P und Q zu haben. Z.B. wenn man ein Modulo $N = P \cdot Q$ in 512 Bits codiert haben möchte, kann man z.B. P in 254 Bits und Q in 258 Bits wählen. Man wird ein Register mit 256 Zellen verwenden für die Berechnungen, die P betreffen, und mit $256 + n = 288$ Zellen für die Berechnungen, die Q betreffen. Bei dem Schaltkreis nach dem Stand der Technik muß man ein Register mit 512 Zellen für die Berechnungen, die Q betreffen, verwenden, was für die Dauer der Berechnungen entscheidend ist.

Adaption von Verfahren an ein gerades Modul.

Der Coprozessor, wie er verwendet wird, erfordert es, daß Modulo N ungerade gewählt wird (das Bit mit dem geringsten Gewicht ist 1). Dies ist eine Notwendigkeit für die Berechnung von J_0.

Der Erfinder schlägt vor, die oben beschriebenen Verfahren auf die Durchführung von modularen Multiplikationen mit einem geraden Modul auszudehnen.

Obgleich dies nicht genau dargelegt wurde, ist es selbstverständlich, daß die modularen Operationen nur dann interessant sind, wenn bei N nicht alle Bits 0 sind.

Wenn N gerade und nicht Null ist, ist es immer möglich, eine binäre Dateneinheit N' zu finden, so daß $N = N' \cdot 2^g$, wobei N' ungerade und $l > g > 1$.

Wenn die nutzbare Größe von N' kleiner als die von N ist, wird man davon ausgehen, daß N' in m' Worten à k Bits codiert ist. Man kann ebenso davon ausgehen, daß die nutzbare reelle Größe von N' (in Worten) m" ist. Diese letzte Option ist selbstverständlich sehr interessant, wenn man die Dauer der modularen Operationen möglichst optimieren will.

Man führt die Operationen durch, indem man beachtet, daß das Modul N' ist, wobei $A \cdot B \cdot 2^{-g} \bmod N'$ erzeugt wird und dieses Ergebnis mit 2^g multipliziert wird (entweder mittels Software in der Zentraleinheit oder durch Verschieben in einem Register).

- Wenn $a + b > m"$:

Das Verfahren wird dadurch durchgeführt, daß:

- $H = 2^{(a+b)\cdot k-g} \bmod N'$ erzeugt wird,

- ein Zwischendatensatz $P(A, H)_{N'} = A \cdot 2^{b\cdot k-g} \bmod N'$ erzeugt wird, codiert in n Bits, indem die m Worte von dem codierten H und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreis angelegt werden, und

- $A \cdot B \cdot 2^{-g} \bmod N'$ erzeugt wird, indem die m Worte des Zwischendatensatzes und die b Worte von B jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden,

- $A \cdot B \bmod N$ durch Verschieben erzeugt wird.

Das letzte Verschieben wird realisiert, je nachdem ob man ein Register oder die Zentraleinheit verwendet.

Es ist selbstverständlich notwendig, das Erzeugen von H ebenfalls zu modifizieren.

Man geht vorzugsweise auf folgende Art und Weise vor:

H1 - Laden von Modulo N' (oder N) in dem Register 12 und Initialisieren des Registers 10, wobei der Inhalt des zweiten Registers B(0) genannt wird ($n = m \cdot k$ Bits auf 0) und Initialisieren des Registers 44 (d.h. Erzeugen eines Signals RES, so daß SC = 0),

H2 - Ausgeben von N' vom Register 12 durch Verschieben nach rechts mit Rückschleifen des Ausgangs an seinen Eingang um l - g Bits (oder um l Bits), wobei l die Zahl von genutzten Bits von Modulo N ist, um ein Bit mit dem größten Gewicht auf 1 in dem Register 12 zu haben,

H3 - Erzeugen und Speichern eines Datenwertes B(1) = 2 · B(0) - N") codiert in n Bits mit $N" = N' \cdot 2^{n-l+g}$, wobei N" dem gespeicherten binären Datenwert in dem Register 12 entspricht, durch:

- Ausgeben von N" und B(0) durch Verschieben nach rechte in den Registern 10 und 12 des Inhalts dieser Register, wobei der Eingang des Registers 10 mit dem Ausgang des Subtraktionsschaltkreises 27 verbunden ist und der Eingang des Registers 12 mit seinem Ausgang verbunden ist,

- Bit-für-Bit-Subtraktion in dem Schaltkreis 27 je nach ihrem Ausgang von Bits von N" und B(0) mit Verschieben nach links um eine Einheit des Ergebnisses, das mit R(0) = B(0) - N" bezeichnet wird,

- Laden des Ergebnisses der Subtraktion nach Verschieben, genannt $B(1) = 2 \cdot R(0)$ in dem Register 10,

- Bit-für-Bit-Subtraktion in dem Subtraktionsschaltkreis 40 von B(1) und N", um festzustellen, ob $B(1) \geq$ oder $< N"$, und Erzeugen durch den Schaltkreis 44 von SC = 0, wenn $B(1) \geq N"$, und von SC = 1, wenn $B(1) < N"$,

H4 - Erzeugen eines Datenwertes $H_{int} = 2^v \bmod N"$ mit $v = n - 1 + m' \cdot k + (a + b - m') \cdot k/2^r$, mit r einer ganzen Zahl, so daß $k/2^r$ eine ganze Zahl ist, durch:

- Durchführen einer Schleife, die mit einem Indes i indiziert ist, wobei i eine ganze Zahl zwischen 1 und v ist, wobei die i-te Interation die folgenden Operationen umfaßt:

- wenn $B(i) < N"$, Laden von $B(i+1) = 2 \cdot (B(i) - 0) = 2 \cdot B(i)$ in das Register 10 nach Verschieben nach links von B(i) um eine Einheit und Bit-für-Bit-Vergleichen von B(i+1) und N",

- andernfalls Bit-für-Bit-Subtraktion von N" und B(i) mit Verschiebung nach links um eine Einheit des Ergebnisses und Laden von $B(i+1) = 2 \cdot (B(i) - N")$ in das Register 10 und Bit-für-Bit-Vergleichen von B(i+1) und N",

H5 - wenn $B(v+1) \geq N"$: Bit-für-Bit-Subtraktion von B(v+1) und N" zu dem Subtraktionsschaltkreis 27, was $2 \cdot (B(v+1) - N"$ ergibt, und Ablegen von $B(v+1) - N"$ in dem Register 10, was ein Verschieben nach rechts um eine Einheit erfordert,

H6 - Verschieben nach rechts um n - 1 Bits in den Registern 10 und 12,

H7 - Durchführen von r P_{field}-Operationen:
$H_{int}(j) = P(H_{int}(j-1), H_{int}(j-1))_{N'}$ mit j Index zwischen 1 und r und $H_{int}(0) = B(v+1) \cdot 2^{l-n}$ oder $(B(v+1) - N") \cdot 2^{l-n}$,

$H_{int}(0) = 2^{v'} \bmod N'$ mit $v' = m' \cdot k + (a + b - m') \cdot k/2^r$.

Man erhält so $2^{(a+b)\cdot k} \bmod N'$.

H8 - Erzeugen des Parameters H durch:

- Erzeugen von $2^{u\cdot k-g} \bmod N'$ mit u einer ganzen Zahl, so daß $u \cdot k \geq g > (u - 1) \cdot k$,

- Durchführen einer Operation P_{field}, um $H = P(2^{u\cdot k-g} \bmod N', 2^{(a+b)\cdot k} \bmod N')_{N'}$ zu erhalten;

- wenn $a + b > m"$ und $B \leq N'$:

ist das Verfahren das folgende:

- man erzeugt $H = 2^{(a+m")\cdot k-g} \bmod N'$,

- man erzeugt einen Zwischendatenwert $P(A, B)_{N'} = A \cdot B \cdot 2^{-a\cdot k} \bmod N'$, codiert in n Bits, indem m Worte, entsprechend B, codiert in m Worten, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikationsschaltkreises angelegt werden, und

- man erzeugt $A \cdot B \cdot 2^{-g} \bmod N'$, indem die m Worte des Zwischendatenwertes und die m" Worte von geringem Gewicht von H jeweils am seriellen Eingang und am parallelen Eingang des Multiplikationsschaltkreises angelegt werden,

- man erzeugt $A \cdot B \bmod N$ durch Verschieben.

Man kann so $H = 2^{(a+m')\cdot k-g} \bmod N'$ erzeugen. Dieses entspricht, die nutzbare Größe von N anstelle der nutzbaren Größe von N' zu betrachten, was nicht vernünftig ist, wenn g so ist, daß $m'' < m'$.

Für die Erzeugung von H bezieht man sich auf die oben beschriebenen Schritte, die einerseits den Fall betreffen, wo $a + b > m''$ mit N gerade, andererseits den Fall, wo $a + b > m'$ und $B \leq N$ und N ungerade. Die Anpassung erfolgt ohne spezielle Probleme für den Fachmann.

- Wenn $a + b \leq m''$:

verfährt man wie folgt:

- Wenn $a + b < m''$ erzeugt man also $A \cdot B \cdot 2^{-g} \bmod N'$, indem m Worte entsprechend $B \cdot 2^{a\cdot k-g}$, codiert in m-Worten, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikationsschaltkreises 19 angelegt werden,

- wenn $a + b = m''$,
dann vergleicht man $B \cdot 2^{a\cdot k-g}$ und N', und

- wenn $B \cdot 2^{a\cdot k-g} < N'$ erzeugt man $A \cdot B \cdot 2^{-g} \bmod N'$, indem m Worte entsprechend $B \cdot 2^{a\cdot k-g}$, codiert in m Worten, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikationsschaltkreises 19 angelegt werden,

- andernfalls erzeugt man $A \cdot B \cdot 2^{-g} \bmod N'$, indem m Worte entsprechend $B \cdot 2^{2\cdot k-g} \bmod N'$, codiert in m Worten, und die a Worte von A jeweils am seriellen Eingang und am

parallelen Eingang des Multiplikationsschaltkreises 19 angelegt werden.

- Man erzeugt A · B mod N durch Verschieben.

Die Erfindung ist insbesondere insoweit vorteilhaft, als man (soweit es die Verfahren betrifft) die existierenden Schaltkreise nach Figur 1 und 2 ohne zu modifizieren verwenden kann. Man modifiziert lediglich den Schaltkreis der Sequenzierung, was das Erzeugen von unterschiedlichen notwendigen Steuersignalen für die Funktion dieser Schaltkreise ermöglicht. Insbesondere ergibt es sich, die verwendbaren Größe des Modulo N und der Operanden zu berücksichtigen. Vorzugsweise wird man die Verfahren der Erfindung anwenden, indem man wie oben angedeutet den Schaltkreis 1 modifiziert.

In Figur 2 wurde davon ausgegangen, daß man die Ressourcen des Schaltkreises nach Figur 2 nutzt. Dies erlaubt es, die Gesamtgröße des Coprozessors zu minimieren. Selbstverständlich kann man auch einen Schaltkreis verwenden, der lediglich zur Berechnung von H ausgelegt ist.

Selbstverständlich ist es klar, daß die durch die Erfindung eingeführten Modifikationen angewendet werden können auf die Realisierung von modularen Quadrierungen und daher modularen Potenzierungen.

EP 0 712 071

Patentansprüche

1. Verfahren zum Durchführen einer modularen Multiplikation nach dem Montgomery-Verfahren, bei welchem ein Multiplikand A und ein Multiplikator B jeweils in a bzw. b Worten à k Bit kodiert sind, wobei die Worte mit dem höchsten Gewicht von A und B nicht Null sind, ein Modulo N kodiert ist in m Worte à k Bit, das Modul (m - m') Worte mit dem höchsten Gewicht à k Bit mit dem Wert Null hat, wobei $0 < m' \leq m$,
umfassend Multiplikationsschritte in einem Multiplikatorschaltkreis (19) mit einem seriellen Eingang, um kodierte Daten zu empfangen, die in wenigstens m' Worten à k Bit kodiert sind, einem parallelen Eingang zum Empfangen von Worten, die in k Bit kodiert sind, und mit einem seriellen Ausgang,
dadurch gekennzeichnet, daß man sukzessive beim Durchführen der Multiplikation eine vorgegebene Anzahl p von Worten am Paralleleingang des Multiplikatorschaltkreises anlegt, wobei p unabhängig von m ist und größer oder gleich der Zahl a ist.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß es einen Schritt umfaßt, bei dem man einen Fehlerkorrekturparameter $H = 2^{p \cdot k} \bmod N$ in m Worten à k Bit kodiert erzeugt.

3. Verfahren nach Anspruch 2, dadurch gekennzeichnet, daß:
   - $H = 2^{(a+b) \cdot k} \bmod N$ erzeugt wird,
   - ein Zwischendatensatz $P(A, H)_N$ erzeugt wird, der kodiert ist in m Worten à k Bit, indem die m Worte von H und die a Worte von A jeweils am seriellen Eingang

und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden, und

- A · B mod N erzeugt wird, indem die m Worte des Zwischendatensatzes und die b Worte von B jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden.

4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, daß es folgende Schritte aufweist:

P1 Erzeugung des Fehlerkorrekturparameters H und eines Parameters J_0, der kodiert ist in k Bit, und Abspeichern von J_0 in einem Register (17) mit k Bit,

P2 Ablegen des Parameters H und von Modulo N in den jeweiligen ersten und zweiten Registern (10, 12) mit n Bit, wobei $n = m \cdot k$, und Initialisieren eines dritten Registers (11) von n Bit auf Null, wobei der Inhalt dieses dritten Registers mit S bezeichnet wird,

P3 Durchführen einer Schleife, die mit i indiziert ist, wobei i eine Variable zwischen 1 und a ist, wobei jede i-te Iteration die folgenden Operationen umfaßt:

a) Übertragung des i-ten Wortes A_{i-1} des Multiplikanden A eines vierten Registers (16) an eine Speicherkippstufe (21),

b) Erzeugung eines Wertes $X(i) = S(i-1) + H \cdot A_{i-1}$ mit $S(0) = 0$ und $S(i-1)$ gleich dem aktualisiert genannten Wert S, der wie nachfolgend definiert ist:

I Verschieben nach rechts des Inhalts des ersten Registers (10) zum Eingang eines ersten seriell-parallelen Multiplikatorschaltkreises (19) mit Rückschleifen des Ausgangs des ersten Registers (10) an seinen Eingang,

II Multiplizieren der Bits von H mit A_{i-1},

III Verschieben nach rechts des Inhalts des zweiten Registers (12) mit Rückschleifung des Ausgangs des Registers an seinen Eingang,

IV Bestimmen des aktualisierten Wertes von S(i-1) als Wert, der in dem dritten Register (11) nach der (i-1)-ten Iteration abgelegt wurde, wenn dieser kleiner als N ist, und durch serielles Subtrahieren von N von diesem, wenn der Wert größer ist, wobei der Ergebniswert der aktualisierte Wert von S(i-1) ist, und

V Verschieben nach rechts des Inhalts des dritten Registers (11) und Addieren des Wertes der Multiplikation $H \cdot A_{i-1}$ Bit für Bit zu dem aktualisierten Wert von S(i-1),

c) Multiplikation des Wortes mit dem geringsten Gewicht von X(i), $X_0(i)$ mit J_0 und Eintragen des Wertes $X_0(i) \cdot J_0 \bmod 2^k = Y_0(i)$ in ein Register (18) und gleichzeitiges Verzögern von N und X(i) um k Zyklen,

d) Berechnen eines Wertes $Z(i) = X(i) + Y_0(i) \cdot N$ durch:

I Multiplizieren von $Y_0(i)$ mit N, verzögert um k Zyklen, und

II Addieren von X(i) zu dem Wert $Y_0(i) \cdot N$,

e) Vernachlässigen des Wortes mit dem geringsten Gewicht von Z(i) und Abspeichern der übrigen Worte, d.h. $Z(i)/2^k$ in dem dritten Register (11),

f) Vergleichen von $Z(i)/2^k$ Bit für Bit mit N, um den aktualisierten Wert S(i) der folgenden Iteration auf die oben beschriebene Art zu bestimmen,

g) wobei das i-te Wort des Multiplikanden A in dem vierten Register (16) zu irgendeinem Zeitpunkt während der obengenannten Operationen abgelegt worden ist;

P4 bei der a-ten Iteration Ignorieren des Wortes mit dem geringsten Gewicht von Z(a) und Eintragen der restlichen Worte in dem ersten Register (10),

P5 Wiederholen der Schritte P3 und P4, bei welchen:

- A durch B ersetzt wird und $Z(a)/2^k$ oder $(Z(a)/2^k) - N$ an den ersten Multiplikatorschaltkreis (19) ausgegeben wird,

- i zwischen 1 und b variiert wird,

P6 Ausgeben des in dem ersten Register (10) gespeicherten Ergebnisses für die letzte Iteration, eventuell nach Abzug von N, falls nötig.

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, daß das Erzeugen des Parameters H in dem ersten Schritt die folgenden Schritte umfaßt:

H1 Ablegen von Modulo N in einem ersten Register (12) mit n Bit, wobei $n = m \cdot k$, und Initialisieren eines zweiten Registers (10) von n Bit auf Null, wobei der Inhalt des zweiten Registers B(0) genannt wird,

H2 Verschieben nach rechts mit Rückschleifung an seinen Eingang von 1 Bit des ersten Registers, wobei l die Zahl der für die Moduloberechnung genutzten Bits ist,

H3 Erzeugung und Abspeichern eines Datenwertes $(B(1) = 2 \cdot (B(0) - N')$, kodiert in n Bit mit $N' = N \cdot 2^{n-1}$, durch:

- Verschieben nach rechts in den zwei Registern und Subtraktion Bit für Bit der Inhalte der ersten und zweiten Register mit Verschieben des Resultats der Bit-für-Bit-Subtraktion, genannt R(0), um eine Einheit nach links,
- Ablegen des Resultats der Subtraktion nach Verschieben, genannt B(1), in dem zweiten Register,
- Vergleichen von B(1) und N',

H4 Erzeugen eines Datenwertes $H_{int} = 2^v \bmod N'$ mit $v = n - l + m' \cdot k + (a + b - m') \cdot k/2^r$ mit r als ganzer Zahl, so daß auch $k/2^r$ eine ganze Zahl ist, durch:

Durchführen einer mit i indizierten Schleife, wobei i eine ganze Zahl zwischen 1 und v ist, wobei jede i-te Iteration die folgenden Operationen umfaßt:

- wenn $B(i) < N'$, dann Ablegen von $B(i+1) = 2 \cdot B(i)$ in dem zweiten Register nach

Verschieben nach links von B(i) um eine Einheit und Bit-für-Bit-Vergleich von B(i+1) und N',
- andernfalls Bit-für-Bit-Subtraktion von N' und B(i) mit Verschieben nach links um eine Einheit des Ergebnisses und Ablegen in dem zweiten Register von $B(i+1) = 2 \cdot (B(i) - N')$ und Bit-für-Bit-Vergleich von B(i+1) und N',

H5 wenn $B(v+1) \geq N'$: Bit-für-Bit-Subtraktion von B(v+1) und N' und Ablegen von $B(v+1) - N'$ in dem zweiten Register,

H6 Verschieben nach rechts um n - 1 Bit in dem ersten und zweiten Register,

H7 Erzeugen des Parameters H durch Durchführen von r P_{field}-Operationen:
$H_{int}(j) = P(H_{int}(j-1), H_{int}(j-1)_N$ mit j Index zwischen 1 und r und $H_{int}(0) = B(v+1) \cdot 2^{1-n}$ oder $(B(v+1) - N') \cdot 2^{1-n}$.

6. Verfahren nach Anspruch 2, dadurch gekennzeichnet, daß B kleiner oder gleich N ist:
- $H = 2^{(a+m')\cdot k} \bmod N$ erzeugt wird,
- ein Zwischendatensatz $P(A, B)_N$ erzeugt wird, der in m Worte à k Bit kodiert ist, indem m Worte, die B in m Worten à k Bit entsprechen, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden, und
- $A \cdot B \bmod N$ erzeugt wird, indem die m Worte des Zwischendatensatzes und die m' Worte mit dem geringsten Gewicht von H jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden.

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß es die folgenden Schritte umfaßt:

P1 Erzeugung des Fehlerkorrekturparameters H und eines Parameters J_0, kodiert in k Bit, und Abspeichern von J_0 in einem Register (17) à k Bit,

P2 Laden des Multiplikators B und von Modulo N in dem jeweiligen ersten und zweiten Register (10, 12) mit n Bit, wobei $n = m \cdot k$, und Initialisieren eines dritten Registers (11) mit n Bit auf Null, wobei der Inhalt dieses dritten Registers S genannt wird,

P3 Durchführen einer mit i indizierten Schleife, wobei i zwischen 1 und a liegt und jede i-te Iteration die folgenden Operationen umfaßt:

a) Übertragung des i-ten Wortes A_{i-1} des Multiplikanden A von einem vierten Register (16) an eine Speicherkippstufe (21),

b) Erzeugen eines Wertes $X(i) = S(i-1) + B \cdot A_{i-1}$ mit $S(0) = 0$ und $S(i-1)$, dem aktualisiert genannten Wert von S, der im folgenden definiert wird, indem:

I Verschieben nach rechts des Inhalts des ersten Registers (10) zum Eingang eines ersten seriell-parallelen Multiplikatorschaltkreises (19) mit Rückschleifen des Ausgangs des ersten Registers (10) an seinen Eingang,

II Multiplizieren der Bits von B mit A_{i-1},

III Verschieben nach rechts des Inhalts des zweiten Registers (12) mit Rückschleifung des Ausgangs des Registers an seinen Eingang,

IV Bestimmen des aktualisierten Wertes von S(i-1) als Wert, der in dem dritten Register (11) nach der (i-1)-ten Iteration abgelegt wurde, wenn dieser kleiner als N ist, und durch serielles Subtrahieren von N von diesem, wenn der Wert größer ist, wobei der Ergebniswert der aktualisierte Wert von S(i-1) ist, und

V Verschieben nach rechts des Inhalts des dritten Registers (11) und Addieren des Wertes der Multiplikation $B \cdot A_{i-1}$ Bit für Bit zu dem aktualisierten Wert von S(i-1),

c) Multiplikation des Wortes mit dem geringsten Gewicht von X(i), $X_0(i)$ mit J_0 und Eintragen des Wertes $X_0(i) \cdot J_0 \bmod 2^k = Y_0(i)$ in ein Register (18)

und gleichzeitiges Verzögern von N und X(i) um k Zyklen,

d) Berechnen eines Wertes $Z(i) = X(i) + Y_0(i) \cdot N$ durch:

I Multiplizieren von $Y_0(i)$ mit N, verzögert um k Zyklen, und

II Addieren von X(i) zu dem Wert $Y_0(i) \cdot N$,

e) Vernachlässigen des Wortes mit dem geringsten Gewicht von Z(i) und Abspeichern der übrigen Worte, d.h. $Z(i)/2^k$ in dem dritten Register (11),

f) Vergleichen von $Z(i)/2^k$ Bit für Bit mit N, um den aktualisierten Wert S(i) der folgenden Iteration auf die oben beschriebene Art zu bestimmen,

g) wobei das i-te Wort des Multiplikanden A abgelegt worden ist in dem vierten Register (16) zu irgendeinem Zeitpunkt während der obengenannten Operationen;

P4 bei der a-ten Iteration Ignorieren des Wortes mit dem geringsten Gewicht von Z(a) und Eintragen der restlichen Worte in dem ersten Register (10);

P5 Wiederholen der Schritte P3 und P4, bei welchen:
- A durch H ersetzt wird und $Z(a)/2^k$ oder $(Z(a)/2^k) - N$ an den ersten Multiplikatorschaltkreis (19) ausgegeben wird,
- i zwischen 1 und b variiert wird,

P6 Ausgeben des in dem ersten Register (10) gespeicherten Ergebnisses für die letzte Iteration, eventuell nach Abzug von N, falls nötig.

8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, daß das Erzeugen des Parameters H in dem ersten Schritt die folgenden Schritte umfaßt:

H1 Ablegen von Modulo N in einem ersten Register (12) mit n Bit, wobei $n = m \cdot k$, und Initialisieren eines zweiten Registers (10) von n Bit auf Null, wobei der Inhalt des zweiten Registers B(0) genannt wird,

H2 Verschieben nach rechts mit Rückschleifung an seinen Eingang von 1 Bit des ersten Registers, wobei l die Zahl der für die Moduloberechnung genutzten Bits ist,

H3 Erzeugung und Abspeichern eines Datenwertes $(B(1) = 2 \cdot (B(0) - N')$, kodiert in n Bit mit $N' = N \cdot 2^{n-1}$, durch:

- Verschieben nach rechts in den zwei Registern und Subtraktion Bit für Bit der Inhalte der ersten und zweiten Register mit Verschieben des Resultats der Bit-für-Bit-Subtraktion, genannt R(0), um eine Einheit nach links,
- Ablegen des Resultats der Subtraktion nach Verschieben, genannt $B(1) = 2 \cdot R(0)$ in dem zweiten Register,
- Vergleichen von B(1) und N',

H4 Erzeugen eines Datenwertes $H_{int} = 2^v \bmod N'$ mit $v = n - l + m' \cdot k + a \cdot k/2^r$ mit r als ganzer Zahl, so daß auch $k/2^r$ eine ganze Zahl ist, durch: Durchführen einer mit i indizierten Schleife, wobei i eine ganze Zahl zwischen 1 und v ist, wobei jede i-te Iteration die folgenden Operationen umfaßt:

- wenn $B(i) < N'$, dann Ablegen von $B(i+1) = 2 \cdot B(i)$ in dem zweiten Register nach Verschieben nach links von B(i) um eine Einheit und Bit-für-Bit-Vergleich von B(i+1) und N',
- andernfalls Bit-für-Bit-Subtraktion von N' und B(i) mit Verschieben nach links um eine Einheit des Ergebnisses und Ablegen in dem zweiten Register von $B(i+1) = 2 \cdot (B(i) - N')$ und Bit-für-Bit-Vergleich von B(i+1) und N',

H5 wenn $B(v+1) \geq N'$: Bit-für-Bit-Subtraktion von B(v+1) und N' und Ablegen von $B(v+1) - N'$ in dem zweiten Register,

H6 Verschieben nach rechts um n - 1 Bit in dem ersten und zweiten Register,

H7 Erzeugen des Parameters H durch Durchführen von r P_{field}-Operationen:

$H_{int}(j) = P(H_{int}(j-1), H_{int}(j-1)_N$ mit j Index zwischen 1 und r und $H_{int}(0) = B(v+1) \cdot 2^{1-n}$ oder $(B(v+1) - N') \cdot 2^{1-n}$.

9. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß man (a + b) und m' vergleicht und
   - wenn a + b < m' dann A · B mod N erzeugt wird, indem m Worte entsprechend $B \cdot 2^{a \cdot k}$ kodiert in m Worte ausgegeben und die a Worte von A jeweils am seriellen Eingang und parallelen Eingang des Multiplikatorschaltkreises angelegt werden,
   - wenn a + b = m', dann $B \cdot 2^{a \cdot k}$ und N verglichen werden, und
   - wenn $B \cdot 2^{a \cdot k} < N$, dann A · B mod N erzeugt wird, indem m Worte entsprechend $B \cdot 2^{a \cdot k}$ kodiert in m Worte und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden,
   - andernfalls A · B mod N erzeugt wird, indem m Worte entsprechend $B \cdot 2^{a \cdot k}$ mod N kodiert in m Worte und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden.

10. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß es einen Schritt umfaßt, bei dem ein Fehlerkorrekturparameter $H = 2^{p \cdot k}$ mod N' erzeugt wird, kodiert in m Worte à k Bit mit N' einem binären Datenwert, so daß $N = N' \cdot 2^g$ und N' ein Bit mit dem geringsten Gewicht hat, das den Wert 1 hat.

11. Verfahren nach Anspruch 10, dadurch gekennzeichnet, daß
   - $H = 2^{(a+b) \cdot k-g}$ mod N' erzeugt wird,
   - ein Zwischendatensatz $P(A, H)_N$, erzeugt wird, kodiert in n Bit, indem die m Worte von dem kodierten H und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden, und

17.09.96

- $P(B, P(A, H)_{N'})_{N'}$ erzeugt wird, indem die m Worte des Zwischendatensatzes und die b Worte von B jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden,
- $A \cdot B \bmod N$ durch Verschieben von $P(B, P(A, H)_{N'})_{N'}$ erzeugt wird.

12. Verfahren nach Anspruch 11, dadurch gekennzeichnet, daß das Erzeugen von dem Parameter H die folgenden Schritte umfaßt:

H1 Ablegen von Modulo N' in einem ersten Register (12) und Initialisieren eines zweiten Registers (10) mit n Bit auf Null, wobei der Inhalt des zweiten Registers B(0) genannt wird,

H2 Verschieben nach rechts des Inhalts des ersten Registers mit Rückschleifen seines Ausgangs an seinen Eingang um l - g Bit, wobei l die Zahl der verwendeten Bits von N ist,

H3 Erzeugen und Abspeichern eines Datenwertes $B(1) = 2 \cdot (B(0) - N")$, kodiert in n Bit, mit $N" = N' \cdot 2^{n-l+g}$, durch:
- Verschieben nach rechts in dem ersten und zweiten Register (10, 12) des Inhalts dieser Register,
- Subtraktion Bit für Bit je nach ihrem Ausgang der Inhalte der genannten zwei Register mit Verschieben nach rechts um eine Einheit des Resultats der Subtraktion, genannt R(0),
- Ablegen des Ergebnisses der Subtraktion nach Verschiebung, genannt B(1) in dem zweiten Register (10),
- Vergleichen von B(1) und N",

H4 Erzeugen eines Datenwertes $H_{int} = 2^v \bmod N"$ mit $v = n - l + m' \cdot k + (a + b - m') \cdot k/2^r$ mit r als ganzer Zahl, so daß $k/2^r$ eine ganze Zahl ist, durch:
Durchführen einer Schleife, die mit einem Index i indiziert ist, wobei i eine ganze Zahl zwischen 1

und v ist, wobei jede i-te Iteration die folgenden Operationen umfaßt:

- wenn $B(i) < N''$, dann Ablegen von $B(i+1) = 2 \cdot B(i)$ in dem zweiten Register nach Verschieben nach links von B(i) um eine Einheit und Bit-für-Bit-Vergleich von B(i+1) und N",
- andernfalls Bit-für-Bit-Subtraktion von N" und B(i) mit Verschieben nach links um eine Einheit des Ergebnisses und Ablegen in dem zweiten Register von $B(i+1) = 2 \cdot (B(i) - N'')$ und Bit-für-Bit-Vergleich von B(i+1) und N",

H5 wenn $B(v+1) \geq N''$: Bit-für-Bit-Subtraktion von B(v+1) und N" und Ablegen von $B(v+1) - N''$ in dem zweiten Register,

H6 Verschieben nach rechts um n - 1 Bit in dem ersten und zweiten Register,

H7 Erzeugen des Parameters H durch Durchführen von r P_{field}-Operationen:
$H_{int}(j) = P(H_{int}(j-1), H_{int}(j-1)_{N'}$ mit j Index zwischen 1 und r und $H_{int}(0) = B(v+1) \cdot 2^{1-n}$ oder $(B(v+1) - N'') \cdot 2^{1-n}$,

H8 Verschieben nach rechts von g Bit von $H_{int}(r)$.

13. Verfahren nach Anspruch 11, dadurch gekennzeichnet, daß für B kleiner oder gleich N':

- $H = 2^{(a+m'')\cdot k-g}$ mod N' mit m" als benötigter Größe an Worten von N' erzeugt wird,
- ein Zwischendatensatz $P(A, B)_{N'}$ in n Bit kodiert erzeugt wird, indem m Worte entsprechend B, kodiert in m Worten, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden, und
- $P(H, P(A, B)_{N'})_{N'}$ erzeugt wird, indem die m Worte des Zwischendatensatzes und die m" Worte von H jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden, und
- $A \cdot B$ mod N durch Verschieben von $P(H, P(A, B)_{N'})_{N'}$ erzeugt wird.

14. Verfahren nach Anspruch 10, dadurch gekennzeichnet, daß man (a + b) und m" mit m" als benötigter Größe in Worten von N' vergleicht und
    - wenn $a + b < m"$ dann $A \cdot B \cdot 2^{-g} \bmod N'$ erzeugt wird, indem m Worte entsprechend $B \cdot 2^{a \cdot k-g}$, kodiert in m Worten, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden,
    - wenn $a + b = m"$, dann $B \cdot 2^{a \cdot k-g}$ und N' verglichen wird, und
    - wenn $B \cdot 2^{a \cdot k-g} < N'$, dann $A \cdot B \cdot 2^{-g} \bmod N'$ erzeugt wird, indem m Worte entsprechend $B \cdot 2^{a \cdot k-g}$, kodiert in m Worte, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden,
    - andernfalls $A \cdot B \cdot 2^{-g} \bmod N'$ erzeugt wird, indem m Worte entsprechend $B \cdot 2^{a \cdot k-g} \bmod N'$, kodiert in m Worte, und die a Worte von A jeweils am seriellen Eingang und am parallelen Eingang des Multiplikatorschaltkreises angelegt werden,
    - $A \cdot B \bmod N$ durch Verschieben der Daten, die gemäß des Vergleichs erzeugt wurden, erzeugt wird.

15. Verfahren nach einem der Ansprüche 1 bis 14, dadurch gekennzeichnet, daß m eine Variable ist.

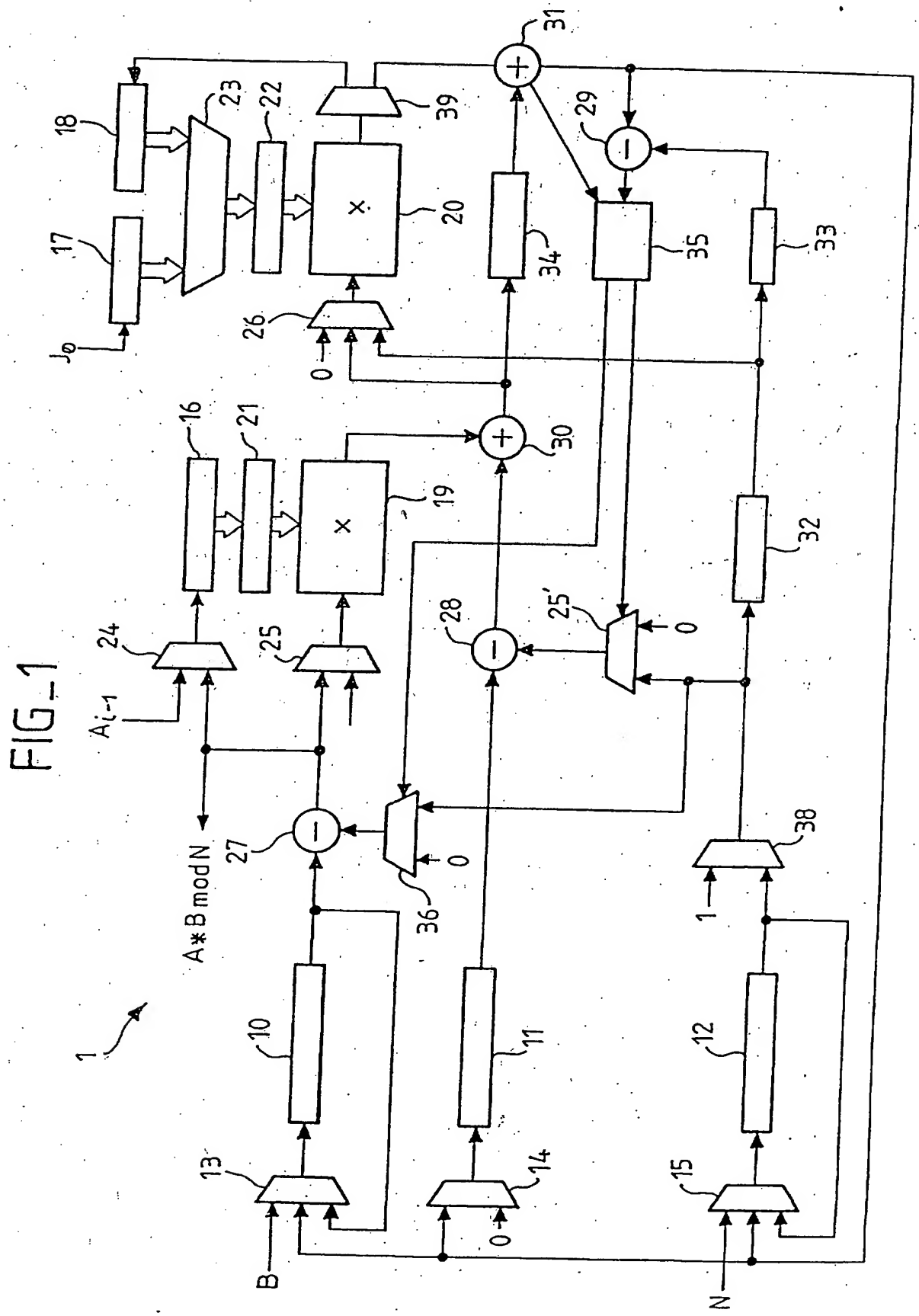
1
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
25'
26
27
28
29
30
31
32
33
34
35
36
38
39
A*B mod N
A_{i-1}
J_0
B
N
0
1
x

FIG_1

17.09.96

191

FIG_2